Application Based Enhancements to WEA.

Mark Wood, CEASa International, England, 2 August 2016.

The opportunity.

The WEA system is based on Cell Broadcast Technology. This has been done because experience has shown that this passive point-to-multipoint bearer service of commercial mobile networks is the most reliable in the presence of severe overload scenarios, such as disasters and terror attacks. In addition, as each cell can be separately addressed, it can passively select the geo scope of messages down to a single cell, about one mile or so in radius.

While this is certainly good enough for most public warning purposes, and is natively and passively available to all mobile devices, there is a problem.

- The design of cells means that the cell footprint, and in particular its borders, do not coincide with political or hazard boundaries.
- Users may get messages which are out of geo scope for their present position, and thus may take location inappropriate advice, or may be irritated by unrelated messaging causing him to disable the feature.
- There may be jurisdictional conflicts as some citizens are receiving advice from an authority other than the one which has sovereignty over the territory in which they currently are.

The Challenge.

We need to enhance the geo fencing of such messages and preserve sovereignty; while still keeping the solution entirely passive. Any super scale messaging would overwhelm conventional one-on-one‘client-server’ solution, and so only a passive solution will protect the system from fatal collapse. CEASa has foreseen this and has appointed ‘Meridian Labs’to propose a solution.

The proposal.

Meridian Labs proposes to CEASa that users may voluntarily participate in a ‘WEA phase 2’ passive high resolution program, by installing an app which can tailor the relevance of any message, down to about 1 meter, and do so passively. This is important so that the very large scale of messaging will not overwhelm either any data channels or any servers, and so any citizen can be assured that he has remained cloaked and that this enhancement in no way changes his privacy rights.

This proposal is that the WEA message is delivered to the phone via cell broadcast, the message is pushed over the existing API (see Appendix B) to an app on the smart phone and checks for an associated ‘polygon’ with this alert message. If there is no polygon, then the message is delivered to the user interface as per the normal standards. If the application finds an associated polygon, it will read the polygon from the cell broadcast and then reverse engineer it into a WGS84 polygon.

The application would then work out its probable location, but only by passive means such as GPS. It would then perform a comparison of the polygon and the location of the device. If there is a match, then the message can be sent to the user interface. If there is not, then the user may choose for himself to see the message or not at his own personal discretion. His local knowledge may be valuable to him in determining the relevance of the message at this moment.

In this way the user will see messages only if he is within the polygon which was defined by the original CAP message, and which has already passed the gateway test of being within the jurisdiction of the sender. Thus the improvements will have been attained.

As all of the processing is done passively and in the terminal, there is no further burden upon the system in doing this improvement. There will be a delay of an estimated 1.8 seconds in the resolution of the polygon, however any terminals not participating will get their message without this delay and thus not be affected at all.

As this process happens passively, there is no requirement for the terminal to report its position without the user's consent, and so no privacy issues can be raised by the use of this app. This both builds trust in the system, and prevents any possibility of abuses which may result in litigation at a later stage.

Further issues to discuss.

Applications do not directly access the underlying hardware and firmware of the mobile device. Rather the application uses a language called 'Application Programming Interface' (API). This in turn signals to the operating system in the device, requesting services from the underlying hardware.

For example; The" Android" operating system is widely used in mobile devices. The applications are created via 'Android Studio', a development environment, and are written in Java. In order for applications to get access to the hardware of the phone they use the Application Programming Interface (API) to communicate with the Operating System of the phone. Thus the applications make API requests to Android to request services, such as Cell Broadcast messages, from the device.

Apple and Microsoft OS devices have similar issues.

The APIs to enable communication between APPs and cell broadcast exist today (See Appendix B and Bibliography below). The problem is that though Cell Broadcast is well defined in the standards 3GPP TS 23.041, and has been for some time, the implementation of its API in smart phones is very variable. In some devices, applications are able to get data from Cell Broadcast, while in others the API is not provided or has been disabled.

In other situations, Engineers have found that the Cell Broadcast API does indeed work, but is not published, so that a good deal of trial and error is needed to work out the APIs. This is completely unacceptable now that Cell Broadcast is well established as a useful bearer service.

Fortunately, Android and other operating systems are frequently upgraded, and so there are opportunities to fix these issues where they are needed. For Example; this has been done recently by Google.

There has been steady progress towards enabling 911 PSAPs to get accurate positions of mobiles who call 911. This is a different situation from mass scale alerting, because the limited scale of such messages does not pose a problem, so active one-on-one technology is appropriate. The announcement below, from Google, announces a change to Android operating systems of API version 9 or higher, (Gingerbread OS), regarding an amendment to the google play location service, so that the location is not sent to google play, but directly to the PSAP (as it not the care for normal google play apps). The following announcement comes from the European Emergency Number Association (EENA).

> *"In one of the biggest news of the industry in the last years, Google announced today that **all Android phones in the world, from Gingerbread OS version onwards, now include Advanced Mobile Location** (AML), an emergency call-based location solution."*
>
> *EENA (2016)*
>
> *Google (2016)*
>
> *http://googlepolicyeurope.blogspot.co.uk/2016/07/helping-emergency-services-find-you.html*

Authorities should engage with Google, Microsoft and Apple to unmask the Cell Broadcast API, so that further enhancements of the emergency alert system can also be made, as it has for the 911 service.

At least the following are needed;

- Be able to receive Cell Broadcast messages including all of their control parameters.
- Be able to open and close Cell Broadcast channels as needed (except presidential).+

Going Forward.

Sustainability is the watchword of our time, and to keep Cell Broadcast alive and useful, it needs to earn its keep. There is a lot of interest in CB for the enhancement of many new services, but it is hampered by the poor implementation of CB's API.

If it were developed properly, it would open a whole new business paradigm which will generate a reliable business revenue stream for commercial mobile networks, as well as solve some pressing technical issues now beginning to arise in the area of "the Internet of Things".

Summary

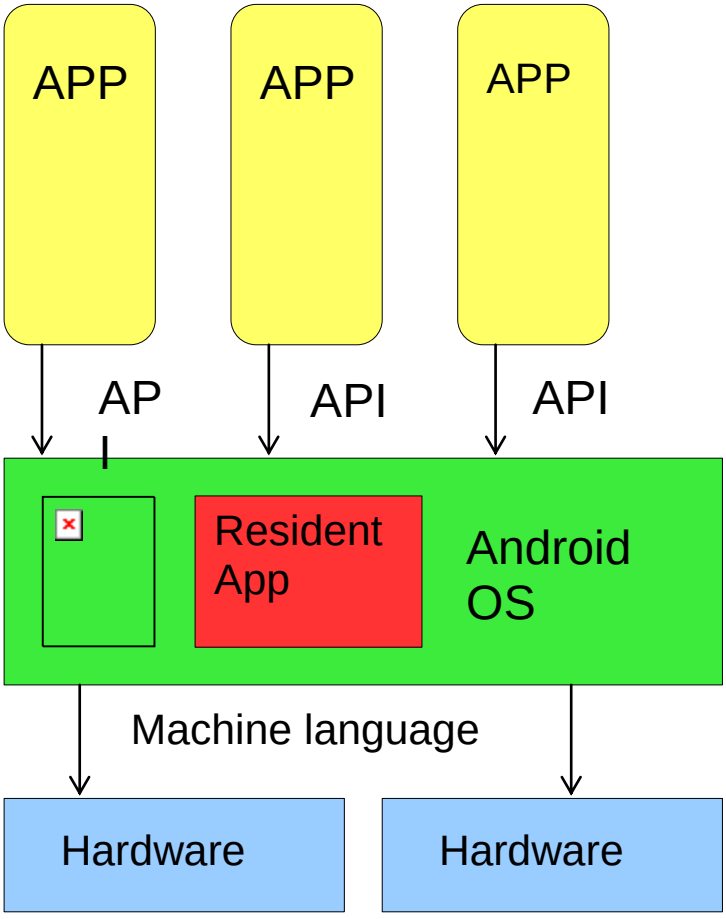
It is technologically very trivial to enhance WEA and cell broadcast in general, by full implementation of the existing standards for CB. However, it will need some engagement with Google Microsoft and Apple, to get the needed changes to the mobile OS done. Meridian labs understand all of this well, and can assist with the technical guidance needed to complete this work, but in order to expedite the changes the initiative must have the support of the wireless industry, public safety and the federal government.

Mark Wood, Secretary CEASa international/Director, Meridian Labs.

mark dot wood at meridianlabs dot org

Appendix A

The diagram below shows the relationship between the Java Applications (APPs) and theAndroid 'Operating System' (OS), which resides in the mobile phone. Applications (APPs) normally get access to the hardware of the phone through the OS via the Application Programming Interface (API). There are API commands to operate CB but they are not published, and in some versions of Android the OS ignores CB commands. However, the CB feature is installed in the phone and can be opened to the APP by simply removing the inhibition. Some applications are resident inside the operating system also have access to the Cell Broadcast feature by direct machine code, thus don't use an API.



Appendix B

An example of some API code used to retrieve a Cell Broadcast message.

Here is an example of an API which calls on the Android Cell Broadcast feature of a Samsung Phone.

```
public class CbReceiver extends BroadcastReceiver {


    @Override
public void onReceive(Context context, Intent intent) {
        //---get the CB message passed in---
        Bundle bundle = intent.getExtras();
SmsCbMessage[] msgs = null;
        String str = "";
if (bundle != null)  {
            //---retrieve the SMS message received---
Object[] pdus = (Object[]) bundle.get("pdus");
msgs = new SmsCbMessage[pdus.length];
for (inti=0; i<msgs.length; i++) {
msgs[i] = SmsCbMessage.createFromPdu((byte[])pdus[i]);
str += "CB lang " + msgs[i].getLanguageCode();
str += " :";
str += msgs[i].getMessageBody().toString();
str += "\n";
            }
            //---display the new CB message---
abortBroadcast();
Toast.makeText(context, str, Toast.LENGTH_SHORT).show();
source; GrAND (2011)
```

See the below link for a discussion about this matter.

http://stackoverflow.com/questions/7118378/how-to-get-cell-broadcast-message

retrieved 25/7/2016.

Bibliography

Android API guide.

https://developer.android.com/guide/index.html

Blog about the challenges of making CB work with some OS.

http://stackoverflow.com/questions/7118378/how-to-get-cell-broadcast-message

Original Cell Broadcast Standard

3GPP 023.041

http://www.3gpp.org/DynaReport/23041.htm

Google announcement regarding upgrade to android to enable location for 911 services.

http://googlepolicyeurope.blogspot.co.uk/2016/07/helping-emergency-services-find-you.html